

**CSC411 A3 Report**  
**Rashid Gaziye**  
**1000607444**  
**gaziyevr**

## 1) 20 Newsgroups predictions

For this task I successfully evaluated K-Nearest-Neighbors, Decision Trees, Logistic Regression, Naive Bayes classifier, and Neural Networks on data converted to tf – idf features. I also tried training SVM but it took too long to compute so I gave up.

K-Nearest-Neighbors and Decision Trees performed the worst among these models. I think this was because the data had many features.

For Logistic Regression I split the dataset into 75% training and 25% test sets. For the parameters I used (tol=0.01, C=2.0, solver='sag'). I noticed that relaxing the regularization by increasing C gave better test performance.

On Naive Bayes model, reducing the smoothing parameter to 0.01 gave me better performance on the test data.

Neural Networks had the best performance but it was close to Logistic Regression and Naive Bayes models. I used logistic activation function with 1 hidden layer of 50 nodes. I noticed that decreasing the momentum had a positive effect on the test error. Activating “early\_stopping” decreased the compute time a lot.

Here's the table of each model's accuracy:

	Train Accuracy	Test Accuracy
Neural Network	93.29%	73.64%
Logistic Regression	92.65%	71.80%
Naive Bayes	95.31%	72.16%
Baseline	59.87%	45.79%

Here is the confusion matrix:

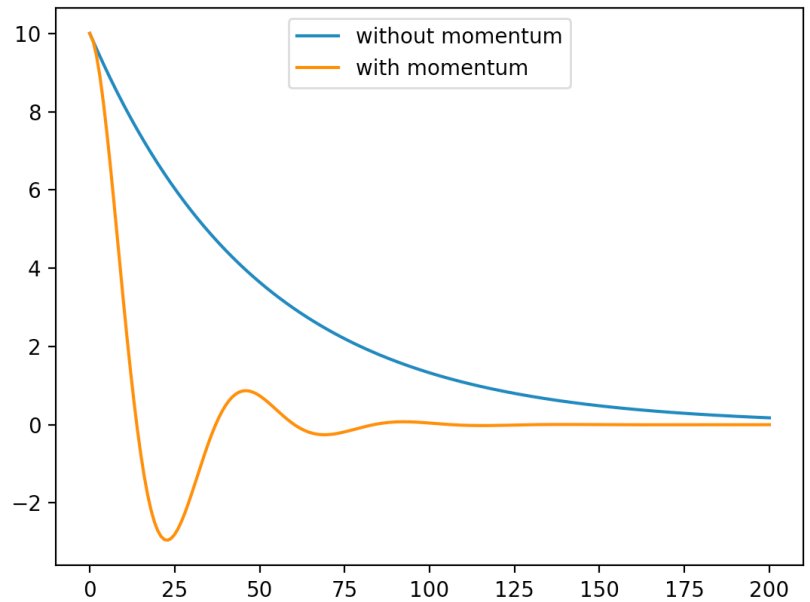
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	115	1	3	0	0	0	0	0	3	3	8	3	2	5	2	15	9	15	16	27	alt.atheism'
2	2	180	16	11	5	18	1	1	2	2	0	5	7	6	8	1	0	1	1	0	comp.graphics'
3	1	8	175	21	6	16	2	0	0	0	0	5	6	0	2	1	2	1	0	3	comp.os.ms-windows.misc'
4	1	4	18	161	10	5	12	0	0	0	0	2	16	0	0	0	0	0	0	0	comp.sys.ibm.pc.hardware'
5	3	6	13	20	159	3	12	0	1	0	0	2	6	0	0	0	1	1	0	0	comp.sys.mac.hardware'
6	1	11	4	1	2	183	0	0	0	0	1	3	0	0	0	0	0	1	0	0	comp.windows.x'
7	0	1	2	5	5	2	201	7	4	2	0	1	4	2	2	0	0	0	0	1	misc.forsale'
8	1	3	2	0	5	0	9	195	13	2	1	2	5	0	3	1	5	0	2	1	rec.autos'
9	5	7	13	6	10	3	11	29	204	12	8	16	14	5	10	8	11	12	6	8	rec.motorcycles'
10	1	3	0	0	0	2	3	1	2	200	6	4	1	0	3	0	2	1	1	4	rec.sport.baseball'
11	1	0	0	0	0	0	1	1	0	6	215	1	0	2	0	0	0	0	2	0	rec.sport.hockey'
12	0	3	0	0	0	2	1	4	0	0	0	183	8	1	1	0	4	1	2	0	sci.crypt'
13	0	8	1	9	12	2	4	4	11	0	0	9	178	5	3	1	1	0	2	1	sci.electronics'
14	2	1	0	0	0	0	0	1	2	2	1	1	4	200	2	0	1	1	4	4	sci.med'
15	7	2	6	1	3	1	1	1	5	0	1	3	5	4	193	4	6	0	4	5	sci.space'
16	33	0	0	0	0	1	2	0	0	1	1	0	0	3	1	211	3	4	0	48	soc.religion.christian'
17	3	0	1	0	1	0	0	2	3	1	3	8	1	1	2	0	165	6	56	13	talk.politics.guns'
18	3	0	1	0	0	0	0	2	2	1	1	0	1	1	0	1	1	187	6	6	talk.politics.mideast'
19	11	0	3	0	0	1	0	2	6	3	0	6	0	6	7	2	15	11	96	8	talk.politics.misc'
20	10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	5	5	0	0	32	talk.religion.misc'

The classifier was most confused about 'talk.politics.guns' and 'talk.politics.misc' with total of 71 cases where it misclassified these classes with each other.

Other than this the classifier also had a hard time differentiating 'soc.religion.christian' vs 'talk.religion.misc' and 'soc.religion.christian' vs 'alt.atheism'

## 2) Training SVM with SGD

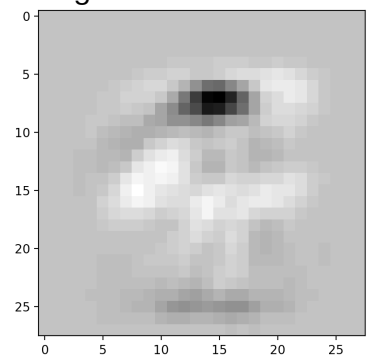
2.1.2: Plot  $w_t$  for 200 time-steps



2.3: Apply on 4-vs-9 digits on MNIST

Training model with momentum 0.0  
mean train loss: 2.98551424402  
mean test loss: 3.16134672986  
train accuracy: 0.8013605442176871  
test accuracy: 0.7983315197678637

Weights for momentum 0.0



Training model with momentum 0.1  
mean train loss: 0.256253868234  
mean test loss: 0.276828322099  
train accuracy: 0.9489342403628118  
test accuracy: 0.9448676097207109

Weights for momentum 0.1

